Fall 12-20-2018

Assessment Of Two Pedagogical Tools For Cybersecurity Education

Pranita Deshpande
University Of New Orleans, pdeshpa1@uno.edu

Assessment Of Two Pedagogical Tools For Cybersecurity Education

A Thesis

Submitted to the Graduate Faculty of the
University Of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science
Information Assurance

by

Pranita Deshpande

B.E. Basaveshwar Engineering College, 2015

December, 2018

This thesis work is wholeheartedly dedicated to my parents,

Mr.Anirudha Deshpande and Mrs. Jyothi Deshpande.

To my sisters Arpita, Ashwita Deshpande and my bother-in-law Vipul Shirodkar

To my nephew Vansh and my best friend Ashwath

Acknowledgment

I would like to thank my advisor, Dr. Irfan Ahmed, for the constant support and being the best guide through out my study. The door to Dr.Ahmed office was always open whenever I ran into a trouble spot or had a question about my research or writing.

I would like to thank Dr. Vassil Roussev and Dr. Minhaz Zibran for offering their time, support and for serving in my thesis Defense Committee.

Finally, I express my profound gratitude to my parents,sisters and my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Figures

List of Tables

Abstract

Cybersecurity is an important strategic areas of computer science, and a difficult discipline to teach effectively. To enhance and provide effective teaching and meaningful learning, we develop and assess two pedagogical tools: Peer instruction, and Concept Maps. Peer instruction teaching methodology has shown promising results in core computer science courses by reducing failure rates and improving student retention in computer science major. Concept maps are well-known technique for improving student-learning experience in class. This thesis document presents the results of implementing and evaluating the peer instruction in a semester-long cybersecurity course, i.e., introduction to computer security. Development and evaluation of concept maps for two cybersecurity courses: SCADA security systems, and digital forensics. We assess the quality of the concept maps using two well-defined techniques: Waterloo rubric, and topological scoring. Results clearly shows that overall concept maps are of high-quality and there is significant improvement in student learning gain during group-discussion.

Keywords: Concept Maps, Peer Instruction, Cybersecurity, Computer Science, Development, Assessment

Chapter 1

Introduction

Cybersecurity is an important strategic areas of Computer Science, and also a difficult discipline to teach effectively. Unfortunately, not much effort has been made to develop course curriculum and instructional material to teach cybersecurity effectively. The goal of this thesis document is to assess the effectiveness of two well-known pedagogical methods/tools for cybersecurity courses i.e., concept maps, and peer instruction.

1.1 Concept Map

In this work we focus on conceptual mapping, which is a well-known pedagogical technique for enhancing students learning and understanding of the concepts [9]. It is graphical tool used by people from many ages which makes easy for one to draw and express their understanding about the topic using the concept map. A concept-map consists of an interconnected elements and sub-elements of a concept, showing a holistic big picture of an overall concept. It is useful to trigger student's active engagement process. In Particular we have developed concept maps for SCADA system security and digital forensics investigation. Having deep conceptual understanding is very important to flourish in these courses. This maps can help the students from diverse background of computer science and engineering to attain in-depth conceptual understanding of about the challenges, issues and solutions of SCADA security and basic foundation for digital evidences and investigation along with varies types of tools and techniques of investigation. To support the use of conceptual mapping in SCADA security and digital forensics class, we develop 22 concept maps for a SCADA security and 19 concept maps for digital forensics course covering different topics from basic to advance levels.

Furthermore, this document assesses the quality of the concept maps using two techniques. A concept maps assessment rubric developed by the University of Waterloo [2] and taxonomy topo-

logical measure defined in Cmapanalysis tool [8]. The rubric suggests to evaluate five elements of a concept map i.e. breadth of net, interconnectedness, use of descriptive links, efficient links,layout and development over the time. It also suggests to assess these elements at four levels i.e. Excellent, Good, Poor, and Fail. Topological taxonomy features suggests the evaluation on the structure of concept map i.e, Branch point count, Average words per concept, concept count, linking phrase, orphan count, proposition count, Root child count, sub map count. Based on these above mentioned parameters it gives the taxonomy score of the concept map.

1.2 Peer Instruction

Peer instruction is a well-defined teaching protocol designed for active engagement of students in class [6, 32]. It involves conceptual multiple-choice questions and group discussion activities aimed to provoke deep conceptual thinking in students. Peer instruction may be effective in dealing with the challenges of cybersecurity education including encouraging out-of-box thinking, developing a mindset of both attacker and defender, and attaining a deep working knowledge of the state-of the art cybersecurity tools and techniques. Inspired by the success of peer instruction in computer science courses, we implement and evaluate peer instruction in a semester-long cybersecurity course, introduction to computer security. Peer instruction requires the students to read lecture material before coming to class. It then, utilizes the acquired knowledge of the students (from the reading) via preplanned conceptual questions to trigger the thinking process in class on a target concept. In a peer instruction classroom, lecture is organized into a set of multiple choice questions. To discuss a concept, instructor first asks a question and then, let the students reply the questions individually followed by a discussion in small groups to resolve any discrepancies in the answers. For this research, we gather the data over three semesters (Fall 2015, Fall 2016 and Fall 2017) consisting of quizzes, subjective exams, peer instruction questions and surveys. The first two semesters are based on traditional lectures while in the latter semester, the course is revised to incorporate peer instruction methodology. This thesis document presents the evaluation results of the implementation and compares them with traditional lecture-centric approach. The peer instruction is evaluated in terms of dropout and failure rates, student learning gain during the group discussion, and survey on students' experience and usage of clickers. The research shows that peer

instruction helps the students in achieving 6% higher grades in final exams than traditional lecture-centric approach. It reduces the failure rate by 61% on average in four core computer science courses (i.e., CS1, CS1.5, Theory of Computation, and Computer Architecture) and improving the student retention in computer science major by 31%. The evaluation results show that peer instruction improves the dropout rate for the undergraduate students by 6% and 16% and the failure rate by 44% and 37% when compared with traditional lecture classes of two semesters respectively. The survey results show that 77% students find the group discussion with fellow students useful to understand the computer security concepts. 70% students would recommend peer instruction be adopted by other instructors.

We have made the concept maps publicly available at gitlab [12]. As the result of this thesis work two papers were published at SIGSCE'19-The 50th ACM Technical Symposium on Computer Science Education conference. Evaluation of Peer Instruction for Cybersecurity Education [14] and Topological Scoring of Concept Maps for Cybersecurity Education [13].

Organization of the Thesis Document Thesis document presents two different works in cybersecurity education. Section 2 discuss the related work on cybersecurity courses and works done on peer instructions and concept maps. Section 3 presents the development and analysis techniques and results on concept maps. Section 4 presents the evaluation methods and evaluation results of peer instruction. Section 5 concludes the thesis document.

Chapter 2

Related Work

2.1 Concept Map

Novak's *el.al*([24], [22], [23])research group at Cornell University first developed concept maps in 1972 in a research project that sought to follow changes in children's understanding of basic science concepts after audio-tutorial instruction in Grades 1 and 2, and continuing through Grade 12. Concept maps were developed to effective improve learning in science. Concept mapping has been shown to be an effective tool for learning at all levels,from preschool to graduate school and corporate training.

Novak *et.al* [9]in there research work they have presented the theoretical foundation of concept maps. They have explained the importance of the concept maps by providing detailed understanding of node and linking phrases to be used. Concept maps are been used world wide and allows learner to organization there thoughts, represent there ideas and knowledge about the topic. Further it also discuss about how it helps instructor to organize the topic and make it easy for students to understand. Thesis document also discuss about how one should construct and use the concept maps.

Instructors grade the students understanding by looking at the concept map developed by them. There are different techniques and challenges in assessing the maps developed.

One method suggested by Novak and Gowin [2] is based on components and structure of cmap. Assessing the map is as follows: valid propositions(1 point each), level of hierarchy(5 points of each level),number of branching(1 point for each branch), cross links(10 points for each cross links), and specific examples (1 point for each example). Here we can see map is graded based on the structure students could develop maps with huge structure but without addressing any conceptual ideas but yet can score good. Addressing the validity and reliability of score became a key issue.

Chen-Chung Liu *et.al* [20] developed a technique to assess the concept map based on the con-

cept included in map a research have suggested assessment based on linkage patterns in concept maps where they propose analytics algorithms for discovering three linkage patterns: Confused concepts,substitute concepts for misconceptions and hidden wrong concept. In this assessment method students concept maps are compared with the experts concept maps for evaluation. Every individual have different way of organizing the knowledge about the specif topics and thus sometimes its difficult to for instructor to assess the maps. The two technologies which we have used to assessment our maps developed are explained further.

Norma L. Miller *et.al* [21] have developed a semantic scoring rubric for concept maps. This rubric can be applied only to the maps that contains some semantic and structural elements to be read meaningfully i.e., concept map with level 3 or greater can be assessed using this scoring method. This method take six criteria: concept relevance and completeness, correct propositional structure, presence of erroneous propositions (misconceptions), presence of dynamic propositions, number and quality of cross-links, and presence of cycles.

Alberto J. Cañas *et.al* and team have a created a cmapanalysis tool for assessing the quality of cmap. They have created a software tool with different types of measurement techniques. This tool measures size, quality and structural evaluation. It is a extensible tool where user can add the measure or techniques they want to use to assess the cmap. Basic Cmap info,Topological Taxonomy Measures, Centrality Measures, Cluster Measures.

Alejandro *et.al* [34] have created a automatic topological taxonomy feature which is used in above explained cmapanalysis tool. This feature classifies the cmaps in 6 level and give results for the structural complexity of the cmap. This Measure uses five description features: the existence of hierarchical structure, size of concept labels, presence of linking phrases, number of branching points, and number of cross links.

2.2 Peer Instruction

Peer instruction is widely adopted and studied in many science disciplines. This section limits the scope to the pertinent efforts on the evaluation of peer instruction.

Crouch *et. al* [11] implement peer instruction methodology on calculus and algebra and present the results of past ten years. They made improvements in implementing the peer instructions such

as replacing class quiz with asking students to read the before-class material and provide a write up in the class.

Rao *et.al* [28] used peer instruction in medical physiology class comprised of 256 first-year medical students. Their implementation is limited to 10 classes. The duration of each class is 50 minutes. Motivation of this research was to improve students performance in quiz by using peer instruction methodology. They used a multiple-choice quiz question followed by class presentation and then, quantified the results. They noticed a significant increase of the percentage of correct answers.

Ronald *et al.* [10] attempted to test the hypothesis that peer instruction enhanced meaningful learning and the student's ability to solve novel problems or the ability to apply the knowledge to different new and existing contexts. They divided a class of 38 undergraduate students into two groups, referred to as group A and group B. The effectivenesses of peer instruction methodology was observed in two exam categories: quiz and problem solving. The group A students followed the peer instruction methodology and were given one-minute time for the group discussions with peer students. The group B students were not allowed to discuss the questions. Their study concluded that peer instruction helped the students in understanding the original material.

Simon *et. al* [33] applied peer instruction methodology to introductory computer science courses. They did not fully-adopt the methodology and deviated from the standard model in some aspects. For instance, textbook reading was assigned before each class and avoided mini lectures before the peer instruction questions and class quizzes. Their evaluation results on quiz and problem solving showed that on average, the correct answers were improved by 21% and 19% after the group discussions respectively.

Johnson *et. al* [16, 17] developed 108 peer instruction questions for digital forensics course and used a subset of these questions in a four-hour long workshop to evaluate the peer instruction methodology. The workshop was attended by 12 participants and covered three digital forensic topics: file system, file carving, and MS Windows registry. Their evaluation results showed the learning gain via quiz and clicker questions by 34% and 13% respectively.

Porter*et. al* [25] present a study on students' failure rate on the data of past 10 years for four computer science courses. Their findings concluded that on average, peer instruction reduced the failure rate by 61% as compared to the Standard lecture-based teaching approach. In particular,

when an instructor teaching the same course using peer instruction, the failure rate reduced by 65%, on average.

Esper *et. al* [15] adopted peer instruction in a software engineering course that had 189 students. They made slight modification in the standard peer instruction methodology. A clicker question is initially shown without answers and then, the instructor asks the students to call out suggestions for the answers. Both the students and instructor proposed a potential answer choices with discussions of those answers. Their survey results showed that 28% students would not recommend peer instruction for teaching because correct answers are not given and clicker questions are not clear.

Daniel it el.al [35] examined the effectiveness of peer instruction in two upper-level computer science courses: Theory of Computation, and Computer Architecture. Their evaluation results found the learning gain of 39% in peer instruction classes.

Chapter 3

Assessment And Development Of Concept Maps

3.1 Developing a Concept Map

Overview Concept maps are a visual tool for organizing and representing knowledge. They include concepts, represented as text boxes, and relationships between pairs of concepts indicated by a connecting link. The most abstract concepts are placed at the top the diagram, while progressively more specific ones are placed underneath them. This simple design allows seamless and effective linking and exploration of concept at different levels of detail. There are many different tools used to create a concept map, the one which we have used is CMap tools. The CMap tools is a software developed at the institute of humans and machine cognition. Computer-based concept mapping software such as CmapTools have further extended the use of concept mapping and greatly enhanced the potential of the tool, facilitating the implementation of a concept map-centered learning environment [9].

3.1.1 Cybersecurity Courses

SCADA System Security Supervisor Control and Data Acquisition (SCADA) systems control major portions of the U.S. critical infrastructure power grid, pipe-lines, water management, etc. and protecting their integrity and availability is of primary importance to national security [5]. Therefore, it is crucial for cybersecurity professionals at-large to have deep conceptual understanding of SCADA security.

SCADA systems are challenging for cybersecurity education because of its interdisciplinary nature and diverse set of applications involving large number of communication protocols, software, and embedded devices [3, 4, 7, 18, 30, 31]. Unfortunately, not much effort has been made to de-

velop course curriculum and instructional material to teach the cybersecurity of SCADA systems effectively.

We develop the concept maps for SCADA system security that can help the students from diverse background of computer science and engineering to attain in-depth conceptual understanding of the challenges, issues and solutions of SCADA security.

Digital Forensics Digital forensics is a challenging discipline to teach effectively because of its inter disciplinary nature. It is defined as the application of scientific tools and methods to identify, collect, and analyze digital artifacts in support of legal proceedings [29]. Students need to accomplish a reasonable critical thinking and understanding of doing digital investigation because they need to have clear understanding of rules and regulations set forth by state and government law [16]. To effective teach digital forensics instructor requires many different practical exercise and examples tasks in class which provokes students with thought-processes and systematically engaging them in problem-solving during class.

3.1.2 Steps to Create a Concept Map

We use the following systematic approach to develop the concept maps for SCADA security and digital forensics course.

1. Select a target concept.
2. Identify keywords that represent some aspect of the concept.
3. Recognize any relationships among the keywords in appropriate words and phrases and then,
4. Draw the concept map; circle the keywords and connect them with the relationship words/phrases.

Guidelines of Do's and Don'ts From our experience of developing and improving concept maps including several revisions, and reviews and comments from other participants, we develop a guideline list of *Do's and Don'ts* while developing a concept map.

- A connection between two nodes should be unidirectional.

- A connecting phrase should describe the relationship between two nodes clearly. Otherwise, avoid such connections and elaborate them with additional keyword(s) between them.
- A connecting loop across one or multiple nodes tend to create confusion and should be avoided.

3.2 Examples of Concept Maps

3.2.1 Concept Maps on SCADA systems

This section presents three examples of concept maps covering three distinct concepts i.e. differences between SCADA and DCS, working of conveyor belt, and attacks on Modbus protocol.

Difference between SCADA and DCS:

SCADA systems are used for a large-scale geographical dispersed physical processes, where as DCS are used to control and monitor the physical processes on a small geographical areas or even single sites.

Steps to develop the Concept Map: Figure 3.1 shows the concept map on the differences between SCADA and DCS (distributed control system). The map consists of four levels of hierarchy, and mostly uses one word to link two nodes. Nodes are self descriptive and mostly contain one or two words. To develop this map, we use our systematic approach as follows:

- We identify the target concept of distinguishing SCADA from DCS.
- We come up with the keywords such as demographic size, reliability/data quality, paradigm, unit design and power consumption as a differentiating parameter between SCADA and DCS.
- The essential connecting words that we used in the map was SCADA and DCS as it connects keywords such as demographic size and what is size if it is a SCADA or DCS.

Working of ICS components: conveyor belt:

The main components of a typical conveyor belt are drivers, actuators, controllers, monitors and sensors. Programmable logic controller (PLC) receives an input signal from proximity sensor that

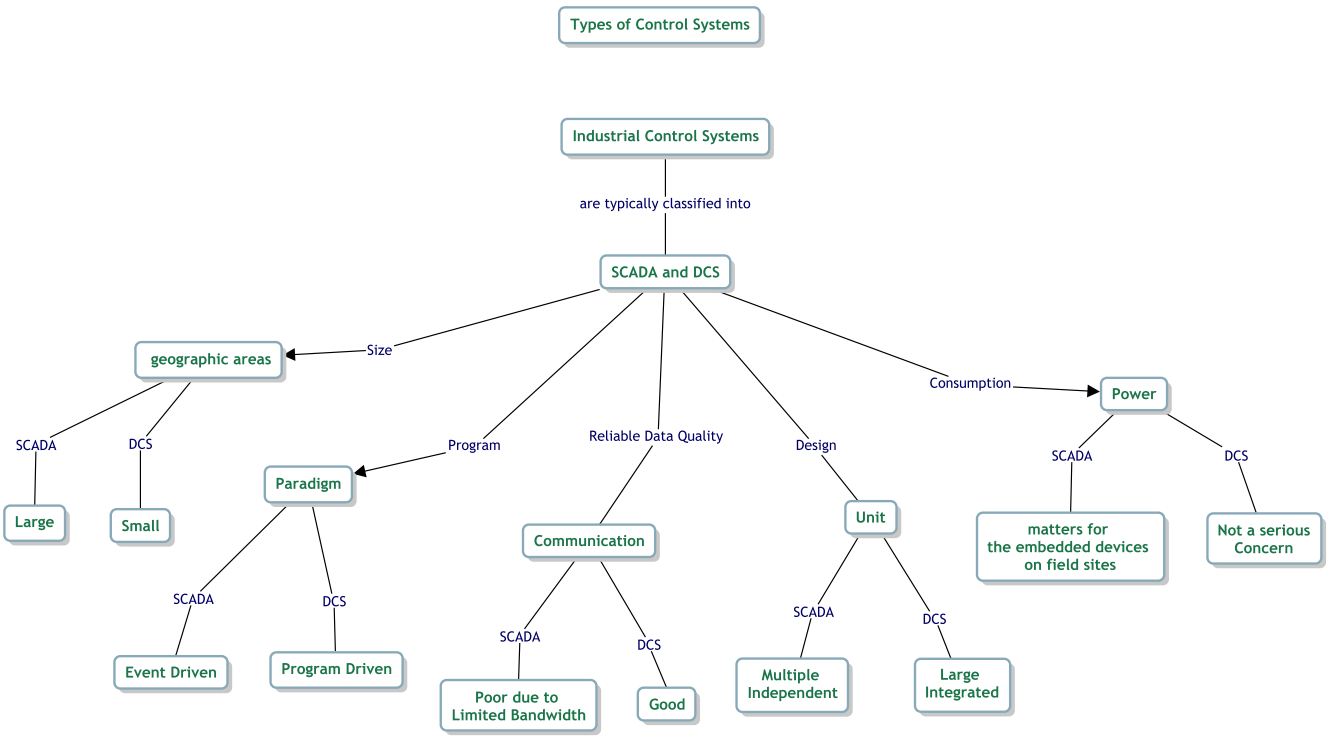


Figure 3.1: Concept map: Differences between SCADA and DCS system

shows that an object is placed on the belt. The PLC runs its control logic and sends an output signal to servo drive to move the conveyor belt to make some space for the next object. The whole conveyor belt physical process can be remotely monitored by using HMI and the data received by the HMI is also stored in historian. There are two types of sensors proximity sensor and photo eye sensor, which detects the presence of the object using beam of light and electromagnetic field respectively.

Steps to develop the Concept Map: Figure 3.2 shows the concept map on the working and components of a conveyor belt. The map consists of four levels of hierarchy, and mostly uses succinct phrases to link two nodes. Nodes are also using short descriptive phrases or long words. To develop this map, we use our systematic approach as follows:

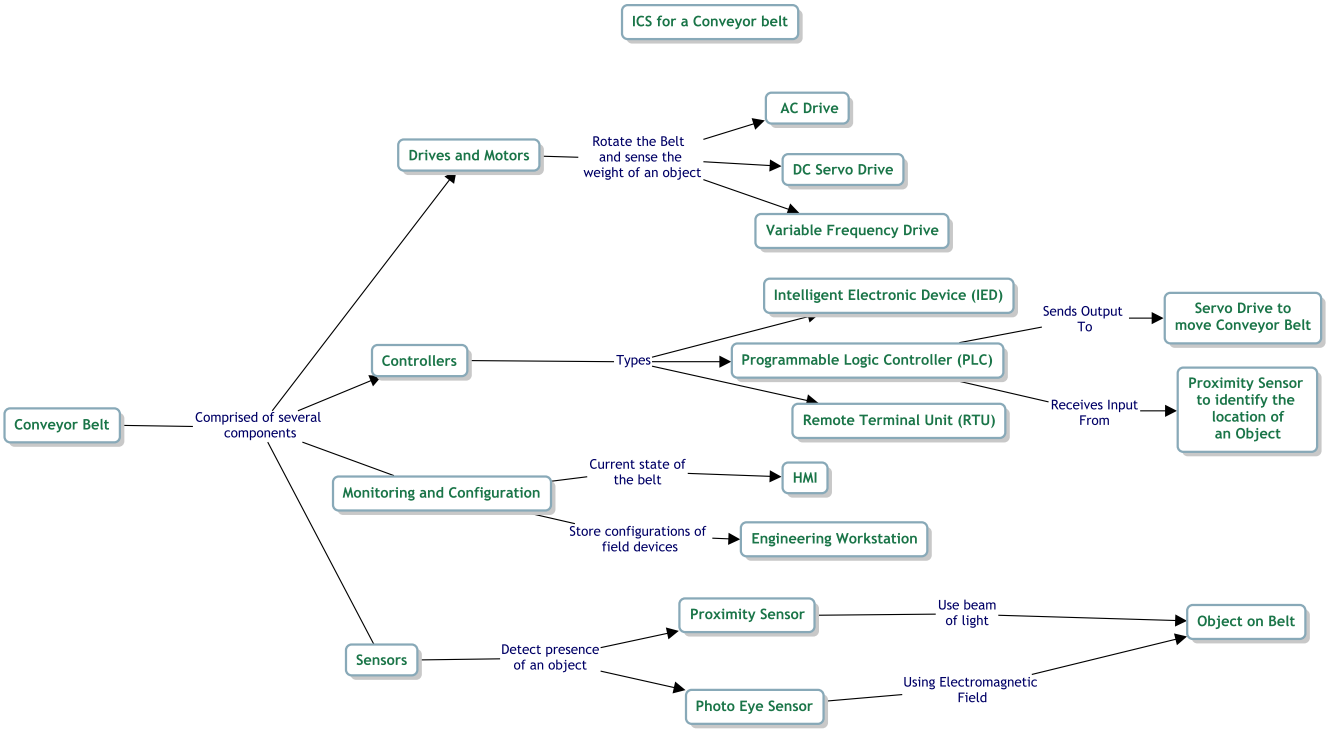


Figure 3.2: Working of ICS: Conveyor Belt example

- The target concept addresses a typical working model of conveyor belt including its components.
- We select the keywords including components, sensors and actuators used and how it was used.
- To connect the nodes that can make sufficient understanding of their relationships, we mostly use phrases, instead of words.

Function codes in MODBUS protocol:

MODBUS is an open communication protocol used for transmitting information between electronic devices. It has various data files and programs files used for reading transmitting data. To read

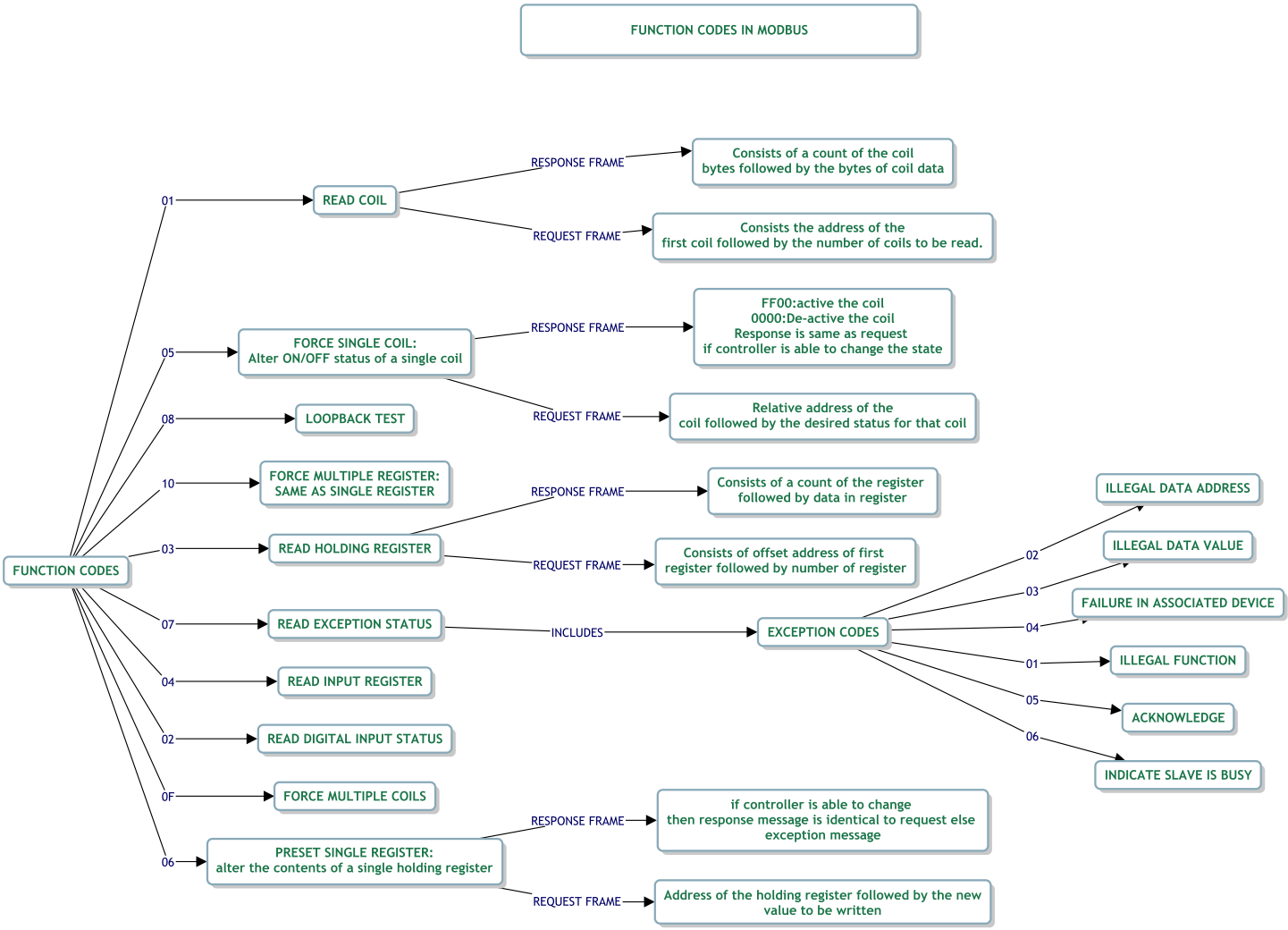


Figure 3.3: Function codes used in MODBUS protocol

transmitted data, we require the use of function codes. Each data file has a function code assigned to read or to write from the concerned data file.

Steps to develop the Concept Map: Figure 3.3 shows concept map explaining function codes of MODBUS protocols. The map consists of four levels of hierarchy. The nodes and connecting links mostly use succinct phrases. To develop this map, we use our systematic approach as follows:

- The target concept is the concept of function codes in MODBUS protocol.
- Keywords are identified Read coil status, Read holding registers and other function codes.
- The connecting words include number of code and what it "indicates" and "message ex-

change".

Real life Attacks on SCADA systems :

There are many different of types of attacks possible on industrial control systems. Students must be aware of these attacks types, causes, consequence and should learn how to prevent or over come some of these attacks. This map shows different categories of attacks and examples of real life incidental. For instance consider the attack which is caused due to delayed SCADA response or where attacker disables the systems and stops systems from reacting. One such example of these attack incident is maroochy waster water attack where attacker steals the equipment, issues radio commands and disable alarms. Consequence was sewage water spillage into local parks and rivers causing environmental damage.

Steps to develop the Concept Map: Figure 3.4 shows concept map explanation different categorizes of real life attack incident on SCADA systems. The map consists of five level of hierarchy nodes and connecting links are self explanatory words. To develop this map, we use our systematic approach as follows:

- The target concept is the concept of providing different categorizes of real life attack incidents
- Keywords are remote SCADA attacks, delayed SCADA response, insider attacks, infecting components, injecting malware and etc.
- The connecting words include attack performance, causes and consequence.

3.2.2 Concept Maps on Digital forensics

This section presents three examples of concept maps covering three distinct topics i.e, handling of the digital evidence, acquisition of evidence and windows registry for investigation.

Handling of digital evidence

Once the evidence is collected from the crime scene protecting the evidence from tampering is very important. Protecting the evidence is equally important steps as of collecting the evidence.

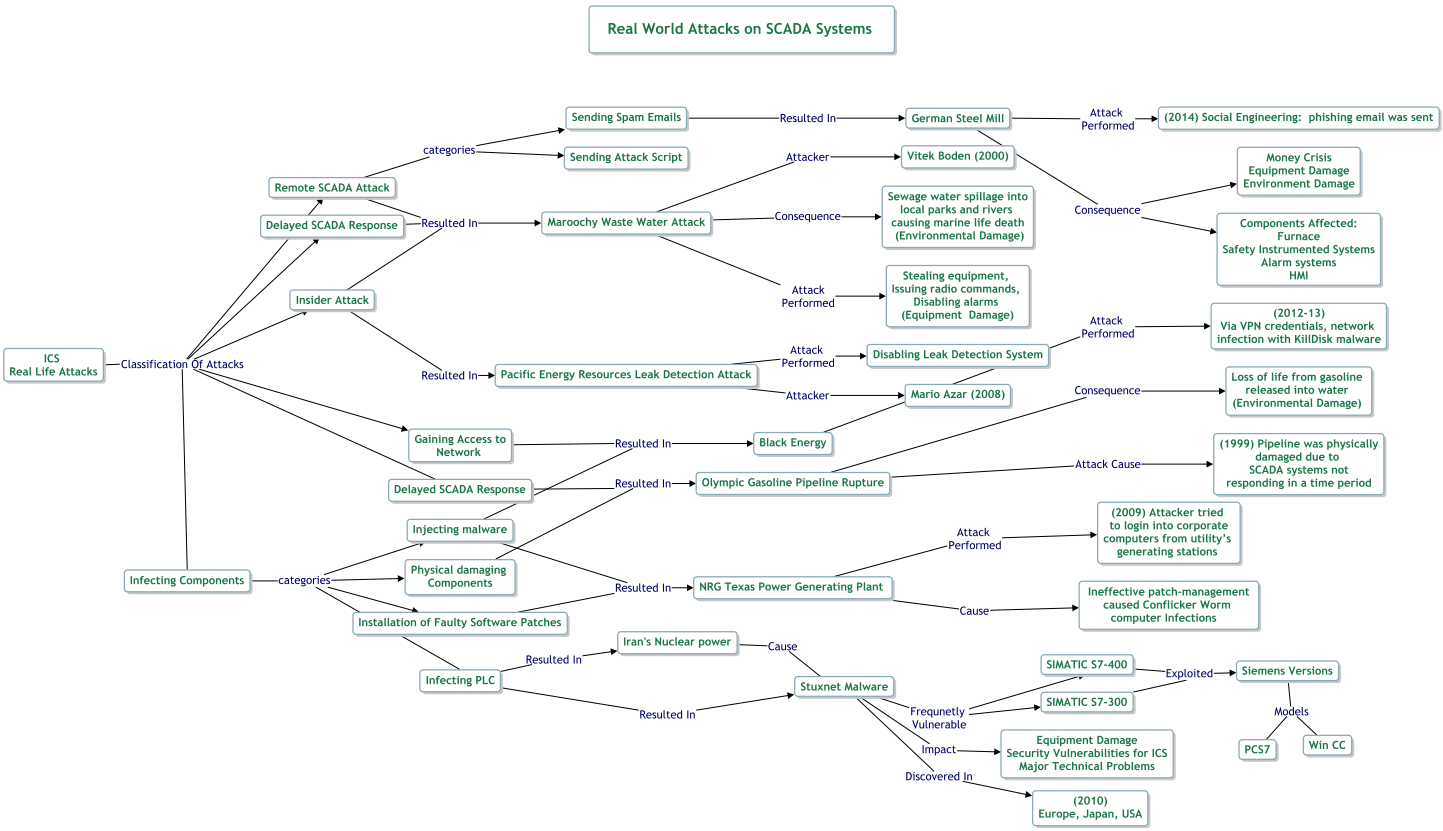


Figure 3.4: Attacks on SCADA systems, Real world attacks examples

Handling of evidence involves five stages i.e, storage of evidence, disposition, transporting, documentation, and packing of evidence. Each step in this process includes further different types of duties to be performed on evidence collected. Storing of evidence means rules and regulation imposed like access to storage must be limited and monitored, chain of custody should be maintained,login and log out details of who, what, when, where and why. Transporting the evidence includes protecting portable devices and media from external corruption, determining if computer should remain powered up,what application were active and other running processes. Where as documenting the evidence requires where the evidence was found, what state it was in, model number,serial numbers and time and date if collection. After the investigation is done evidence must

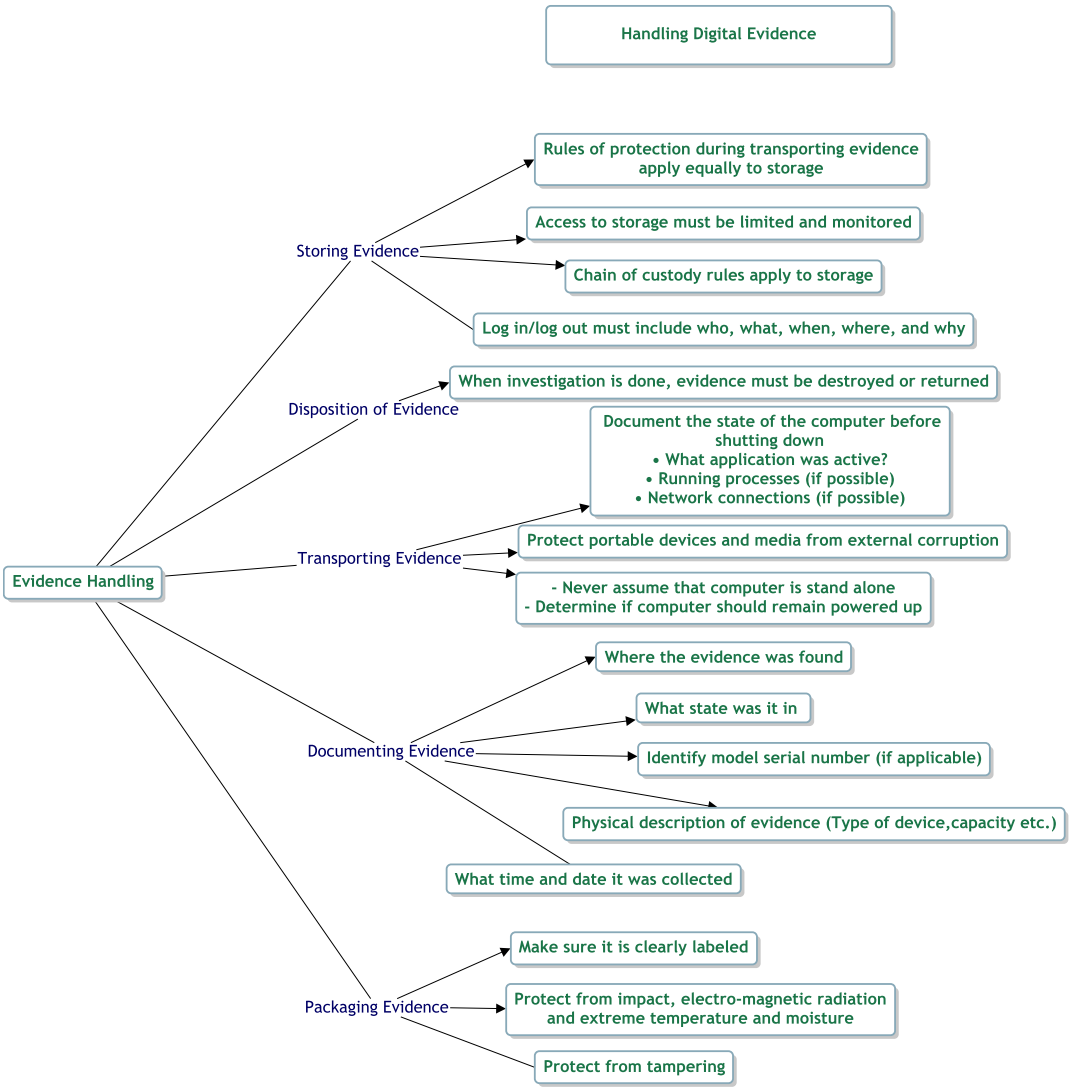


Figure 3.5: Different stages of handling a digital evidence

be destroyed or returned.

Steps to develop the Concept Map: Figure 3.5 shows the concept map explaining in detail the stages included in handling of evidence. The map consists of three levels of hierarchy. The nodes and connecting links are self explanatory phrases. To develop this map we uses our systematic approach as follows:

- The targeted concept is the concept of handling of digital evidence.
- Key nodes indicated the actions and duties to be performed in the each individual stages
- connecting nodes indicated the different stages in digital evidence handling

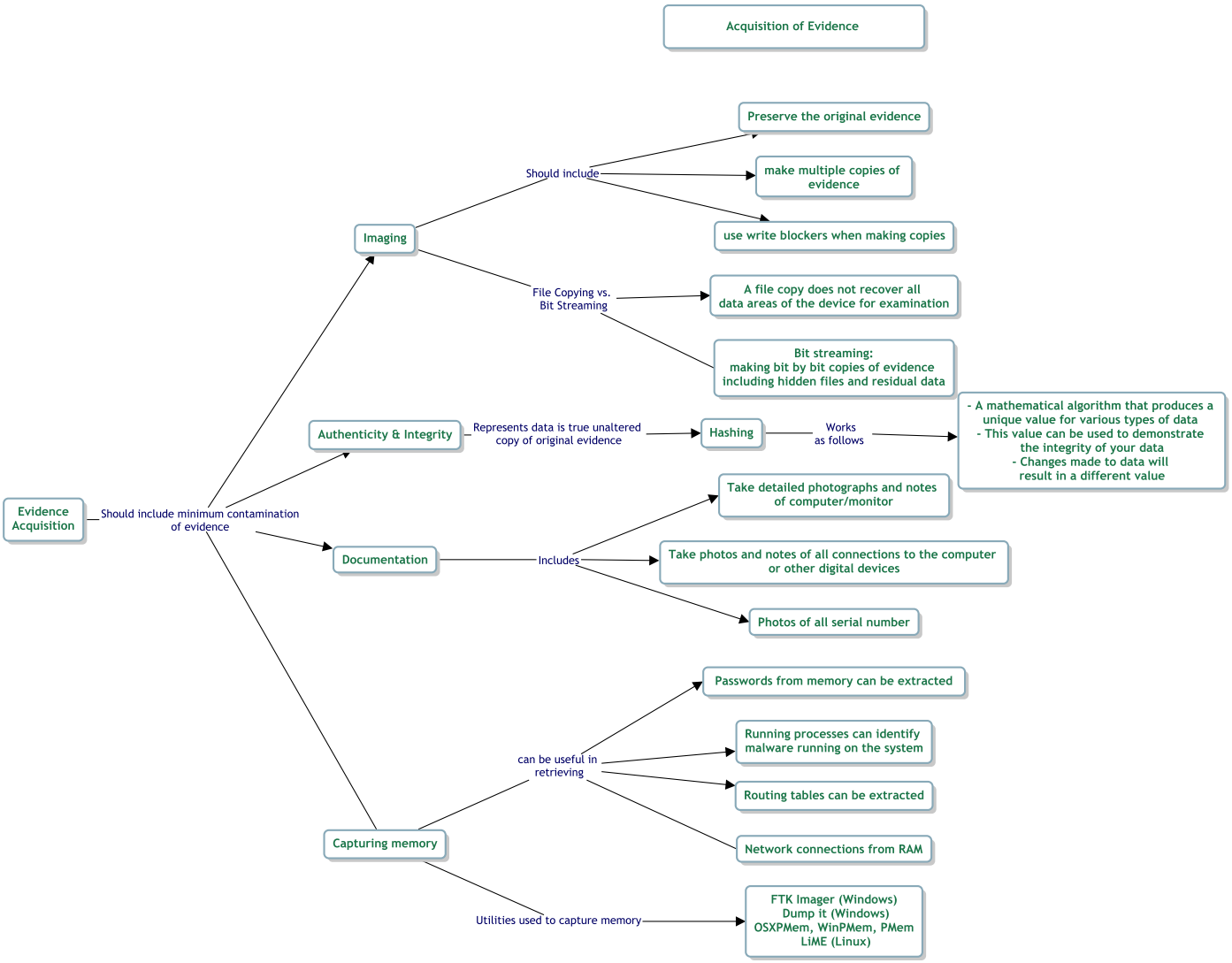


Figure 3.6: Different stages in Acquisition of evidence

Acquisition of evidence

In Investigation steps acquisition of the evidence is the first step performed by investigator. Acquisition means collection of digital evidence. This steps includes different methods of providing the authenticity and integrity of the evidence collected. For instance imaging of the evidence should be done to preserve original evidence. Usage of write blockers when making copies, file coping and bit streaming while imaging. Where as documentation includes taking multiple photos and notes of computer/monitor, crime scene, all devices connected to computer, photos of serial and model numbers of devices.

Steps to develop the Concept Map: Figure 3.6 shows the concept map about different phases of acquisition of evidence. This map consists of four level of hierarchy. The nodes and connecting links are self explanatory phrases. To develop this map we have used our systematic approach as follows:

- The targeted concept is the concept of acquisition of evidence
- Key nodes indicated different steps included in collection of digital evidence
- connecting nodes indicated process included in each steps and what it "indicates"

Windows Registry

While investigating windows registry is a place where investigator get most of the details of system information, recent activities performed on the systems and information related to the systems users. There are five main registry hives to store in information on windows machine and they are security, systems, software, SAM and default. For example system hive contains the information regarding, computer name, list of USB storage devices, list of printers, CPU information, time zone information etc. Software hive contains windows version information, last login users details, recent documents viewed, typed URL's, list of installed applications etc.

Steps to develop the Concept Map:

Figure 3.7 shows the concept map about windows registry hives and windows registry investigation. This map is two level of hierarchy. The nodes and connecting links are self explanatory phrases. To develop this map we have used our systematic approach as follows:

- The targeted concept is the concept of windows register investigation
- Key nodes are "keys" to find specific detail of the computer for instance active computer name is system hive
- connecting nodes indicates the hives and database details of windows registry.

3.3 Concept Maps Assessment methods

As mentioned earlier we analyzed the concept maps using two different techniques.

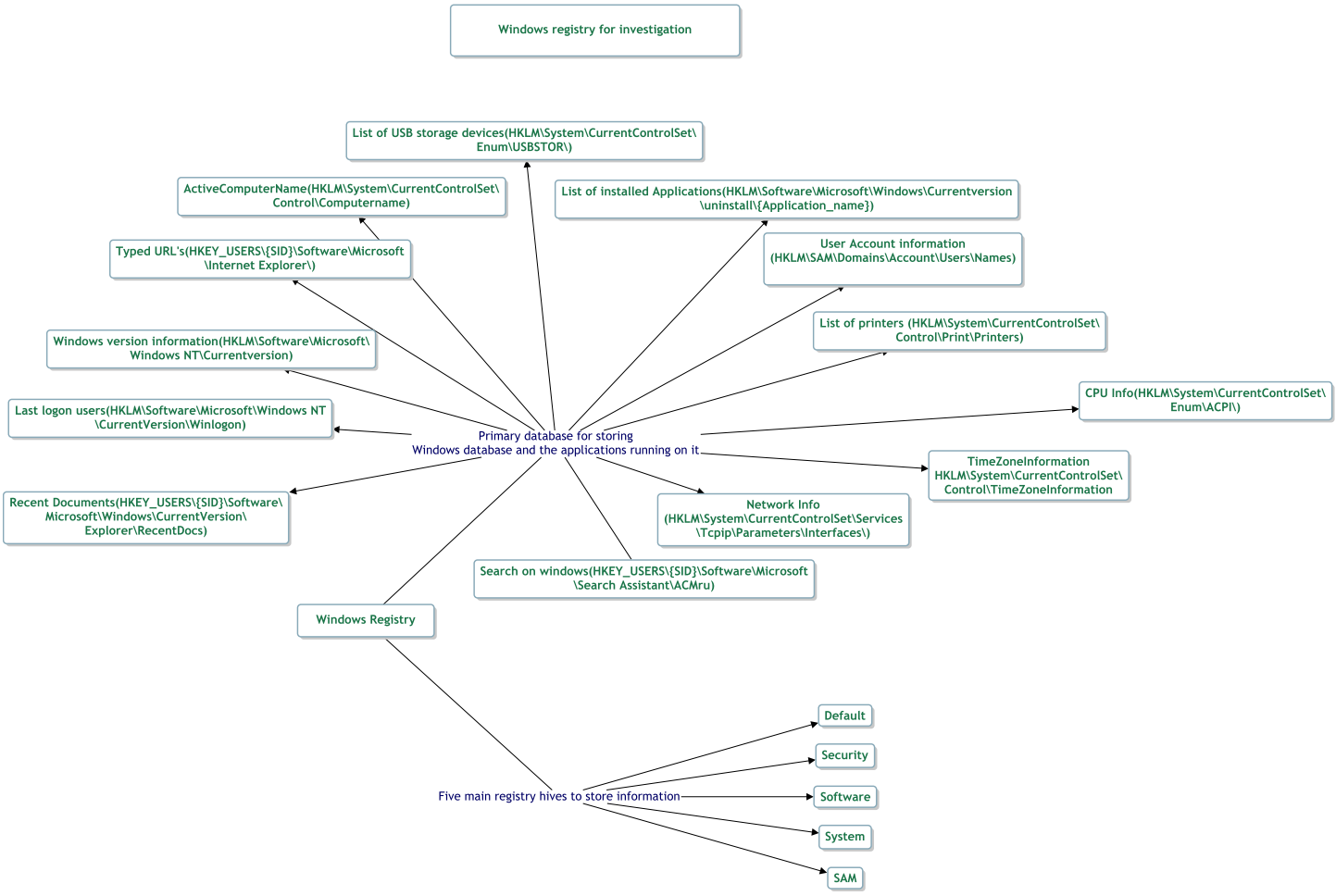


Figure 3.7: Windows registry investigation

3.3.1 Waterloo Rubric

To assess the context in the concept maps developed we have used the Rubrics (developed by University of Waterloo) designed for the assessment of the concept maps [2]. The rubric presents six elements of assessment of a concept map i.e. *breadth of net*, *interconnectedness*, *use of descriptive links*, *efficient links*, *layout* and *Development over time*. It also suggests to assess these elements at four levels i.e. *Excellent*, *Good*, *Poor*, and *Fail*.

3.3.2 Assessment Parameters

Breadth of net: evaluates the significance of target concepts and their description in multiple levels. For *excellent*, a map includes important concepts and describe them in multiple levels. However, for *fail*, a map misses many important concepts.

Interconnectedness: evaluates the number of concepts interlinked with other concepts. For *excellent*, all concepts are interlinked, and for *fail*, few concepts are interlinked.

Use of descriptive links: evaluates the quality of description as accurately defined to vague and incorrectly defined. The first is ranked as *excellent* while the later is *fail*.

Efficient links: evaluates the uniqueness of the information of the links and the quality of description of the relationships among the nodes. For *excellent*, each link type is distinct and clearly describes the relationship, while for *fail*, most links are vaguely described, and not distinct from each other.

Layout: evaluates the physical layout of a concept map including its size to be fit in one page, and hierarchical structure. For *excellent*, maps fit in one page and have clear multiple hierarchy, while for *fail*, map consists of multiple pages and has no hierarchical organization.

Development over time: evaluates whether a concept map is built incrementally as the term progress and new concepts are learned. for *excellent*, final map shows considerable cognitive progression from base map and a significantly greater depth of understanding of the domain. while for *fail* final map shows no significant cognitive profession from the base map and no increase in the understanding of the domain.

3.3.3 Assessment Scoring

As mentioned earlier rubric suggest assessment of six important features of concept maps. The scoring range from 0-4, 0 being the failing concept maps and score for excellent concept maps. Lets consider and example and go through the process of scoring the map. Figure 3.5 show different stages of handling the evidence collect. concept maps are manually scored by using rubric.

Breadth of net: Maps explains all the important ways of handling the evidence. For example it covers storage of evidence, transportation of evidence, documentation, packing and disposition of digital evidence. It also explains in details the about the above methods. This parameters score of 4 since the descriptions are in multiple levels

Interconnectedness from figure we can see that all the nodes/concepts are interlinked, all nodes explains a way to handle the evidence. for instance sub concept transposition of evidence describes the ways to transport the evidence, like should protect the portable device and and media from external corruption ans so on. This parameters gets score of 4 since all the concepts are interlinked.

Use of descriptive links in this maps all the links are self explanatory words making it easy to understand phrases like "storing the evidence" means how the evidence collected must be stores how the evidence should be transported ans so on . Thus this parameters gets a score of 4.

Efficient links all the links and nodes are unique and linking phrases clearly explains the relation of main concept with subconcept. From figure we can see all the nodes/linking phrases are unique, this parameter gets a score of 4.

Layout we can clearly see that concept maps easy fits into a paper and there are no cycles. All the nodes are clearly pointing to their subconcept node. This parameter gets score a of 4.

Development over time this parameter explains how well the map can be extended when further work is done or how easy new subconcept can be added in the cmap. We can see since we have a sub node of a subconcept it is easy to ass any number of subconcept under the main concept. Thus this parameter gets a score of 4.

After the above explanation we can see that all the parameters gets a rank of 4 and thus this concept maps can be scores as excellent quality CMap.

3.3.4 Topological Scoring

To assess the structure of concept map developed we have used topological taxonomy measurement in cmapanalysis tool. Topological taxonomy measures include the taxonomy score between 0 to 6 where higher score typically indicated higher quality concept maps and higher structural complexity. This measure also includes the following individual aspects of the concept map that are considered in calculating the taxonomy score [8]. Cmapanalyis tool can be downloaded from the git respiratory [1].

3.3.5 Assessment Parameters

Average Words per Concept: The total count of words, as separated by whitespace, in all concepts divided by the number of concept in the map. Concise concepts are important to the taxonomy score.

Branch Point Count: The total number of concepts and linking phrases that have at least one incoming connection and more than one outgoing connection.

Concept Count: The number of concept in the map.

Linking Phrase Count: The number of linking phrases in the map.

Orphan Count: The number of concepts in the map that have no connections.

Proposition Count: The number of propositions (i.e. concept-linking phrase-concept) in the map.

Root Child Count: The number of concepts in the map that have an incoming connection from a root concept. A root concept is defined as one that has outgoing connections but no incoming connections.

Sub Map Count: The number of root concepts found in the map.

3.3.6 Assessment Scoring

From the above parameters a taxonomy score is computed for the camps. Score indicated that level 1 cmaps are unevaluated that means maps doesn't meet the minimum requirement criteria to be read meaningfully. Rank 2-3 indicated very low level of concept map, level 4 is intermediate and level 5-6 are high level concept maps. This score is calculated by considered above mentioned parameters.

CMapAnalysis tool runs the classifier in the following procedure: it iterates through the map starting from level 0 and check if map belongs to level N if yes then check for level N+1 until the

highest level is meet. If the condition fails at any level then the map will be classified as N-1 level map. For instance, for a map at level three following conditions are required. *No long concept labels*, *No linking-phrases missing*, *At least 3 branching points* and *Less than 3 hierarchy levels* . Scoring calculating formulas: *concept,labelsize(c) < 12* , *linking phrase, label size(l) >0*, *branching ponit(m) ≥ 3* here c represent concepts, l represents linking phrases of concept map M

3.4 Analysis of Concept Maps

3.4.1 SCADA Concept Map Assessments

We have developed 22 concept maps for the SCADA security course work. Topics for the concept map are included from introductory to advance level. These 22 maps developed are divided for 5 different course modules. The distribution of concept maps with respect to their topics are presented in Table 3.1.

- *Introduction to SCADA Systems* covers the basic concepts of a SCADA system, and its components, provides a brief understanding of some physical processes.
- *Programming of the Programmable Logic Controllers (PLC)* mainly covers Ladder Logic programming including rules to write a program and addressing formats of PLC
- *SCADA communication protocols* covers two protocols, MODBUS and DNP3 along with there header and message formats.
- *SCADA Vulnerabilities and Attack* covers real-world attacks and vulnerabilities discussed in research document along with attack taxonomies in MODBUS and DNP3 protocols.
- *SCADA security solutions* covers security solution in SCADA systems like PLC code detection, smart grid for power stations and smart city application and challenges.

Rubric Results:

Figure 3.8 shows the assessment results we got by using the rubric mentioned earlier, for the concept maps developed for SCADA security course. From the figure we can see that for the

Topics	# of Concept Maps
Introduction to SCADA Systems	4
PLC Programming	3
SCADA communication protocols	6
SCADA Vulnerabilities and Attack	5
SCADA security solutions	4
TOTAL	22

Table 3.1: Number of concept maps developed for different SCADA security topics

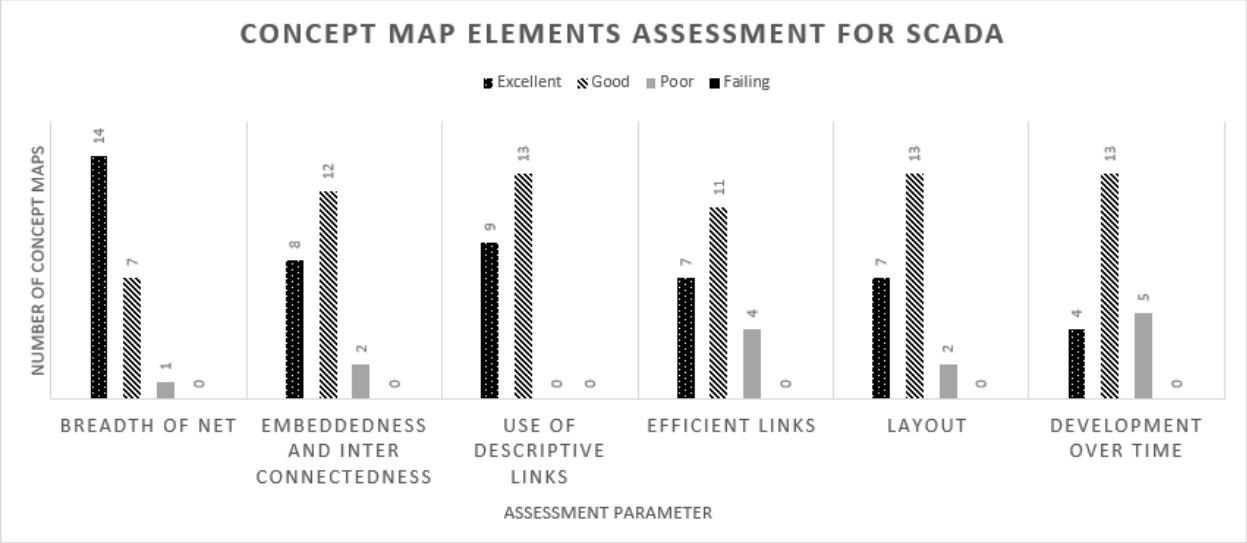


Figure 3.8: Assessment results for SCADA concept maps using *Rubric*

parameter *breadth of net* which says about important concept included in the map and description of domain on multiple level 14 maps are graded to excellent where as 7 maps are good and 1 maps is poor. For *Interconnectedness* 8 map are excellent, 12 maps are good and 2 maps is poor. For *use of descriptive links* 9 maps are excellent and 13 maps are good. For effective links 7 maps are excellent , 11 maps are good, 4 maps are failing. For *layout* 7 maps are excellent 11 maps are good and 4 maps, 4 maps are poor and so on.

Topological Taxonomy Results:

Figure 3.9 shows the results of assessing the concept map by using cmapanalysis tool again the topological taxonomy measure. As mentioned earlier this analysis gives a score for structure of concept map from 0-6, higher the score is means higher quality of concept map. From the figure we can see that most of the maps have a higher rank in topological taxonomy score. out of 22 maps 8 maps score a rank of 2 and below. where as 14 maps score 3 and above in which 3 maps have

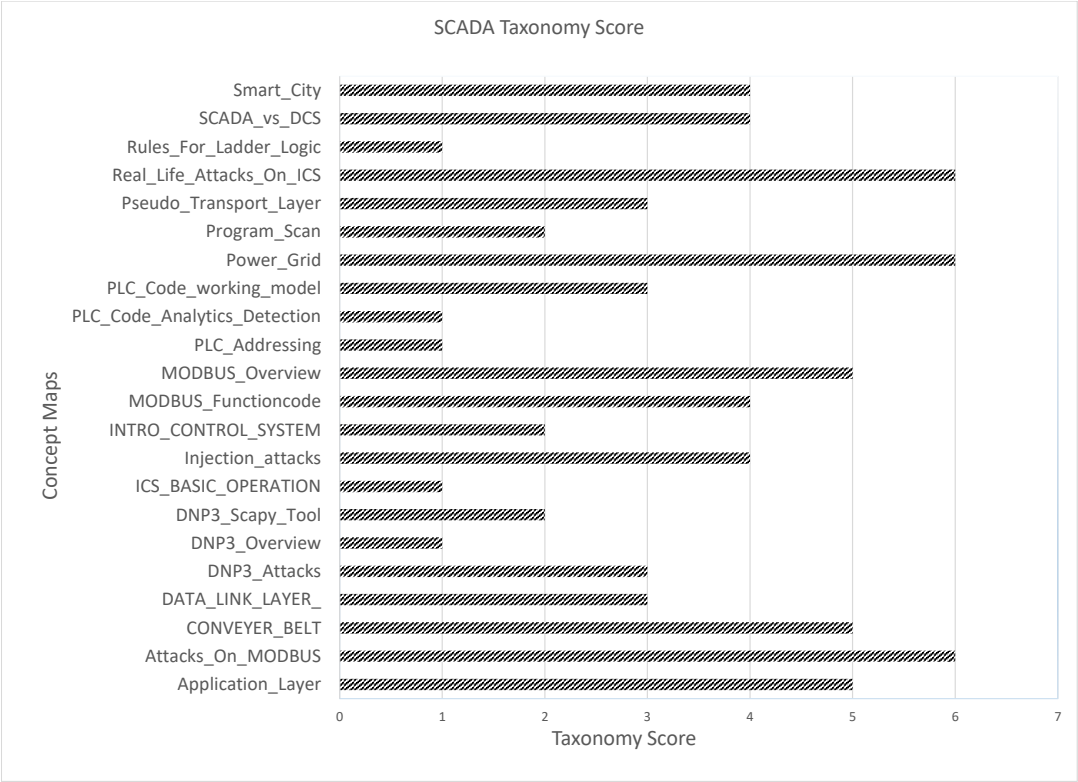


Figure 3.9: Assessment results for SCADA concept maps using *Topological taxonomy*

higher rank i.e., sixth and 3 maps score fifth rank.

Rubric evaluation results show that the concept maps comply with the evaluation criteria and mostly obtain the level of *excellent* where as in topological taxonomy features evaluation most of the concept maps and obtain a score of *4-6*.

3.4.2 Digital Forensics Concept Map Assessment

We have developed 19 concept maps for digital forensics investigation course work, topics for the concept maps are included from the introductory topic of investigation to the advance level like tool used in investigation and file systems. These 19 maps developed are divided into 6 different course modules. The distribution of the concept maps with respect to their topics are presented in Table 3.2.

Topics	# of Concept Maps
Introduction to digital forensics	4
First response and evidence handling	2
Investigation steps	3
File systems	5
Memory Forensics	2
Tools for investigation	3
TOTAL	19

Table 3.2: Number of concept maps developed for different digital forensics topics

- *Introduction to digital forensics* covers concept maps on digital evidence including the where evidence can be found and types of evidence. Documentation of digital evidence, types of digital forensics investigation and legal aspects considered for investigating.

- *First response and evidence handling* covers concept map on how a digital forensics investigator should respond to a case before starting the investigation, what are the necessary steps and procedures which should be taken care of and how the evidence should be handled.

- *Investigation steps* this concept map focus on the steps/tasks that should be performed during a forensic investigation including the acquisition and analysis of the evidence, and the reporting that describes the entire investigation procedure and give a conclusion on a case.

- *File systems* concept map on file system investigation. It provides an overview of different file system, file allocation table, new technology file system, and investigating tips and techniques on file system.

- *Memory Forensics* covers concept maps on memory analysis and live forensics. It explains volatility data and how important the data is for investigation. These maps also discusses volatility framework, which is a popular for investigating volatile data. It explains the volatility plug-ins.

- *Tools for investigation* concept maps on usage of different tools and techniques for file system investigation including sleuth kit, windows registry and web browser investigation.

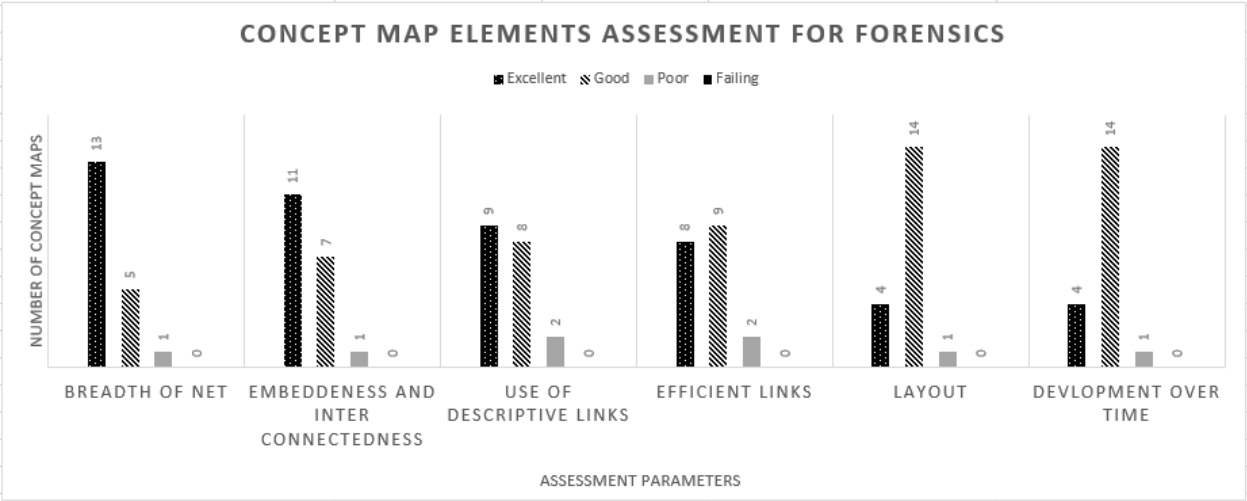


Figure 3.10: Assessment results for Digital forensics concept maps using *Rubric*

Rubric Results:

Figure 3.10 shows the assessment results we got by using the rubric, for the concept maps developed for digital forensics course. From the figure we can see that for *breadth of net* 13 maps are excellent, 5 maps are good and 1 maps is poor. For *Interconnectedness* 11 maps are excellent, 7 maps are good and 1 maps is poor. For *use of descriptive links* 9 maps are excellent, 8 maps are good and 2 maps are poor. For *efficient links* 8 maps are excellent, 9 maps are good and 2 maps are poor. For *layout* 4 maps are excellent, 14 maps are good, 1 map is poor and so on.

Topological Taxonomy Results:

Figure 3.11 shows the results of assessing the concept map by using cmapanalysis tool again the topological taxonomy measure. As mentioned earlier this analysis gives a score for structure of concept map from 0-6, higher the score is means higher quality of concept map. From the figure we can see that most of the maps have a average rank in topological taxonomy score. Out of 19 maps 9 maps score a rank of 2 and above. Where as other 10 maps have a score of 1. Highest rank for digital forensics concept maps is 4 which is for the topic of "report witting of investigation".

Rubric evaluation results show that the concept maps comply with the evaluation criteria and mostly obtain the level of *excellent* where as in topological taxonomy features evaluation most of the concept maps and obtain a score of *2-5*.

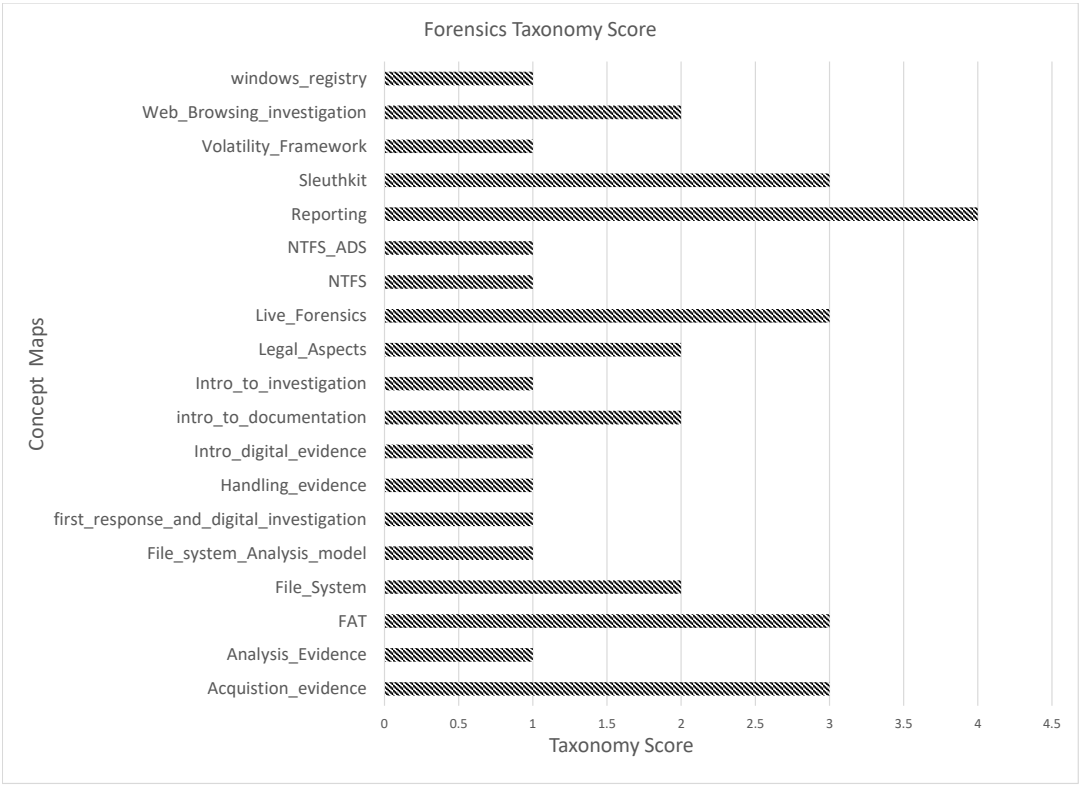


Figure 3.11: Assessment results for Digital forensics concept maps using *Topological taxonomy*

3.4.3 Waterloo Rubric Vs. Topological taxonomy

As described earlier we have used two assessment methods for evaluating the concept maps developed. Rubric which defines the correctness and quality of concept maps and topological taxonomy feature of CmapAnalysis tool which gives the structural level of concept map. Using the rubric as ground truth we have evaluated the accurately the topological taxonomy feature. Figure 3.12 and 3.13 shows the comparison between the ground truth rubric and topological taxonomy feature score. We can see from 3.12 for concept maps on attacks on MODBUS, Power grid and real life attacks both the scoring method evaluate these concept maps as excellent quality maps. For maps on DNP overview, DNP3 scapy tool results are opposite and where as for other maps we have mixed results. For figure 3.13 we can clearly see that there is no 100 % accurate result. Further evaluation on comparison results is needed to see the accuracy level of topological taxonomy feature.

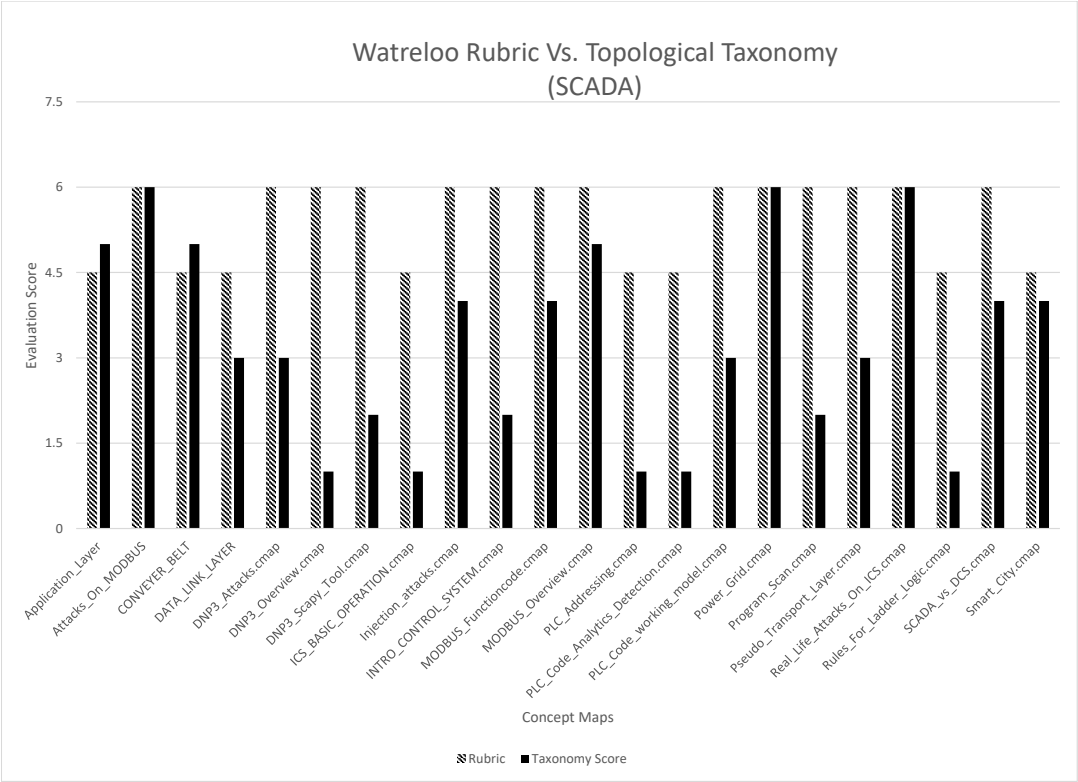


Figure 3.12: Comparison results for SCADA

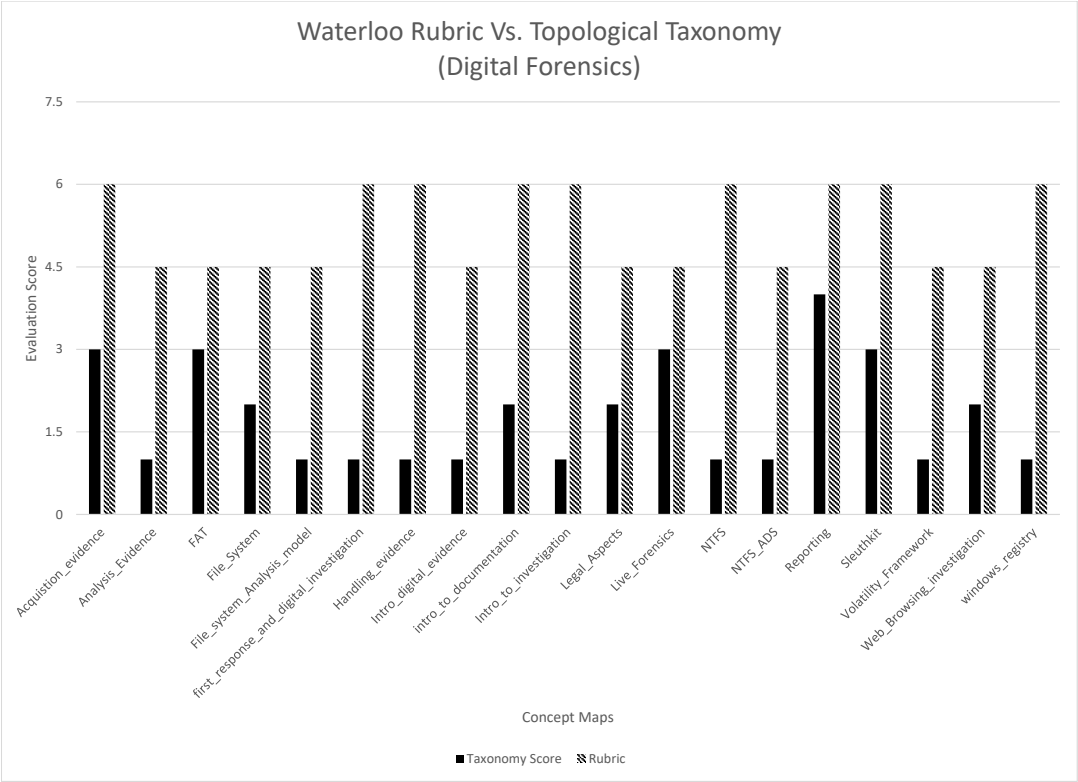


Figure 3.13: Comparison results for Digital Forensics

Chapter 4

Evaluation of Peer Instruction

Peer instruction is a teaching protocol consisting of before-class and in-class activities.

Before class Before-class activities are aimed to have students familiar with the lecture topics of next class. They include reading material, online videos etc. A quiz is given with the activities to incentive the students to complete the work.

In class Instructor divides a lecture into a series of peer instruction questions. Each questions targets a certain concept. The lecture begins with a question. The instructor typically provides 60-90 seconds to the students to respond to the question and then, allows the students to discuss their answers with fellow students in small groups. The discussion typically lasts for two to three minutes. After the discussion, the instructor presents the same question to the students to respond. Clickers are used to collect the individual responses of the students immediately on instructor's computer that summarizes the results. If the answers are incorrect, the instructor may further choose to discuss the concept, otherwise, can pose the next question.

4.1 Peer Instruction Implementation

Course We choose the *introduction to computer security* course to evaluate the effectiveness of peer instruction methodology for cybersecurity education. The course is taught at both undergraduate and graduate levels and provides a broad overview of cybersecurity and covers at least four cybersecurity areas i.e., user authentication, malicious software, buffer overflow, and cryptographic tools. The course is offered regularly once or twice in a year as needed.

Instructor The course instructor is an experienced teacher who taught several cybersecurity graduate and undergraduate courses. He taught the *introduction to computer security* course five

Table 4.1: Number of students enrolled in the introduction to computer security course.

Semesters	Undergraduate	Graduate	Total
Fall 2015	6	17	23
Fall 2016	13	6	19
Fall 2017	23	10	33

times before implementing the peer instruction in the course. His student evaluations are typically around 4.5 out of 5.0, which validate the high quality of instruction.

Peer Instruction Activities Recall that peer instruction teaching involves before-class and in-class activities. For the implementation, the students are given reading assignments to cover before-class activities. Each assignment expects the students to read a book chapter of the topic discussed next week in class. The students are given at least one-week time to finish an assignment.

During class, the instructor asks peer instruction questions on a target topic and let students discuss their answers in small groups (consisting of typically four to five students). Clickers are used to collect the responses. The instructor also used his lecture slides to discuss the topics as needed.

4.2 Data Collection

To assess the effectiveness of peer instruction in terms of student failure rate and learning gain, we develop and utilize four different instruments for data collection i.e., Quiz, Subjective Exam, Clicker Questions and Survey (refer to Table 4.2 for a summary). Figure 4.1 shows the timeline of data collection activities in a semester. The semester starts with a before-class reading assignment. The students are given a week to complete it while the instructor uses this week to discuss the syllabus, introduce the course activities, go through hands-on assignments, and initiate discussion on computer security to raise the students' interest on the subject matter. The rest of the semester comprises of periodic reading assignments and data collection activities.

We have collected the data for three semesters i.e., Fall 2015, Fall 2016, and Fall 2017. The first two uses traditional lecture approach while the latter implements peer instruction. Table 4.1 shows the enrollment number of undergraduate and graduate students for these semesters. Unlike

Table 4.2: Data collection instruments

Quiz Questions	Subjective Exams	Clicker Questions	Survey Questions
29	17	18	19

Table 4.3: Survey on the reasons for enrolling in computer security

Survey Question	Fall 2015	Fall 2016	Fall 2017
Interested in subject matter	96%	90%	88%
Times of class is favorable for schedule	28%	16%	27%
Other classes wanted were full	9%	26%	30%
Prerequisite for other classes	13%	37%	30%

graduate student population, the undergraduate enrollment increases over the semesters. This section further describes the data collection instruments.

Quiz Three quizzes are developed to assess the student knowledge on three topics i.e., computer security overview, user authentication, and cryptographic tools. The students are given (at least a week) time to prepare for the quizzes after the lectures on the respective topics are completed in class. The quiz questions are designed to be straight forward with correct set of choices. To quantify the student responses, each correct question is given one mark.

Subjective Exam The exams are midterm and final tests consisting of subjective questions to evaluate the understanding of the students on five cybersecurity topics, i.e., computer security overview, buffer overflow, user authentication, malicious software, and cryptographic tools. The duration of an exam is one hour and fifteen minutes. The students are advised to provide direct and concise answers to the questions. The graduate students are expected to answer one additional question within the allotted time to comply with the university rule. A standard rubric of correct answers is used to quantify the level of understanding of students on the topics.

Clicker Questions The clicker questions are the peer instruction questions used for the lecture in class. Clickers are used to record the polls of a question before and after the student discussion in small groups. The polling results of the questions are an effective means to measure the learning gains of students at micro-scale as a result of peer discussion. Eighteen questions are used for five

Table 4.4: Survey on students background and interest in computer security

Survey Question	Fall 2015	Fall 2016	Fall 2017
Previously taken any coursework related to computer security	22%	16%	28%
Intend to specialize in computer security field	48%	63%	49%
Intend to take additional computer security course after this class	70%	74%	64%

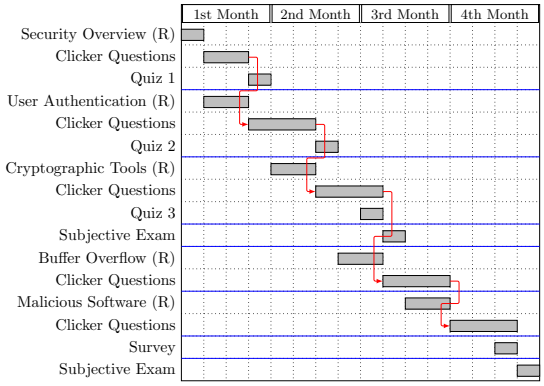


Figure 4.1: Timeline of the data collection using quizzes, survey, subject exams, and clicker questions. Each box represents a week. 'R' identifies before-class reading activities on five topics.

topics. Unfortunately, we could not collect the peer instruction data on one topic i.e., security overview. The other data is collected, analyzed and presented in this document.

Surveys We utilize an attitudinal survey to record the students' experience and opinions on clickers and peer instructions. The survey instrument is provided by Beth Simon and Leo Porter of UC San Diego, and Cynthia Lee of Stanford University. Results from this survey instrument have been published for numerous peer instruction[1] courses, providing useful comparisons for our evaluation of peer instruction for cybersecurity (*e.g.*, [19] [26] [27]).

The survey gathers information on prior usage of clickers, course preparation, peer discussion, clicker usage, and lecture pacing. It contains 19 questions that are designed with a Likert scale. The survey is given to students at the end of semester in class and provided ample time to complete.

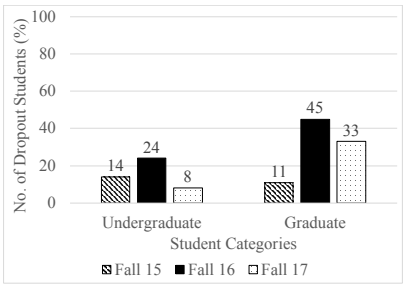
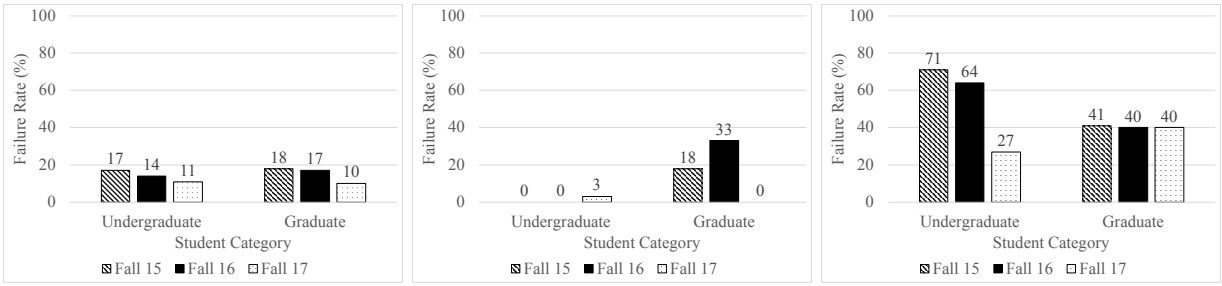


Figure 4.2: Student dropout rate for peer instruction (Fall 17) and traditional lecture (Fall 16 and Fall 15)

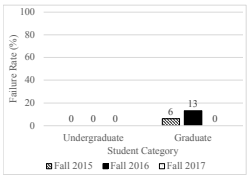


(a) Security Overview (b) User Authentication (c) Cryptographic Tools

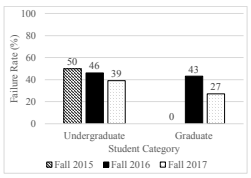
Figure 4.3: Failure rate in *quizzes* for peer instruction (Fall 17) and traditional lecture (Fall 16 and Fall 15) in three cybersecurity topics i.e., security overview, user authentication, and cryptographic tools at both graduate and undergraduate levels.

Table 4.5: Student Survey on Peer instruction lecture preparation, peer instruction, and clicker usage

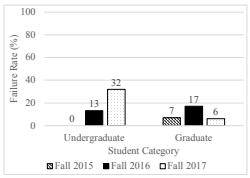
Survey Questions	**Average Opinion**
Thinking about clicker questions on my own, before discussing with people around me, helped me learn course material.	70%
I read The required material before the lectures.	60%
Most of the time my group actually discussed the clicker question.	87%
Discussing course topics with my seatmate in the class helped me better understand the course material	77%
The immediate feedback from the clickers helped me focus on weakness in my understanding of the course	77%
Knowing the right answer is the only important part of the clicker question.	30%
Generally, by the time we finished with a question and discussion, I felt pretty clear about it.	80%
Clickers are an easy-to-use class collaboration tool.	77%
Clickers helped me pay attention in the class compared to traditional lectures	73%
Using clickers with discussion is valuable for my learning.	67%
I recommend that other instructors use this approach (reading quizzes, clickers, in-class discussion) in their courses.	70%



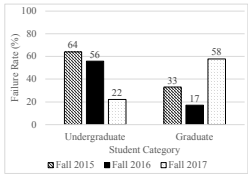
(a) Security Overview



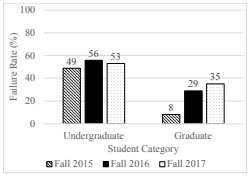
(b) User Authentication



(c) Cryptographic Tools



(d) Buffer Overflow



(e) Malicious Software

Figure 4.4: Failure rate in the *subjective exam* for five topics i.e., introductory computer security, user authentication, and cryptographic tools) at both graduate and undergraduate levels in the peer instruction class (Fall 17) and traditional lecture class (Fall 16 and Fall 15)

4.3 Data Analysis

We analyze the data to measure the effectiveness of peer instruction in terms of dropout and failure rates, student learning gain during group discussions, and students' experience on clicker usage and peer instruction teaching methodology. Tables 4.3 and 4.4 summarize the students' background and interest in computer security. Only 30% students have some prior understanding of cybersecurity. However, 96% students are interested to learn cybersecurity. Around 75% students intend to take more cybersecurity courses and 50% would specialize in this area.

4.3.1 Dropout Rate

At the university, the students may drop the course within two weeks after the semester starts without any official record. After two weeks, the students have six weeks to drop the course with a "W" (or Withdraw) grade recorded.

Figure 4.2 shows the dropout rate of the undergraduate students for both traditional lecture classes conducted in Fall 2015 and Fall 2016, and peer instruction classes in Fall 2017. We notice that the dropout rate is reduced by 6% and 16% in peer instruction classes at undergraduate level if compared with Fall 2015 and Fall 2016 respectively. However, the dropout rate for graduate students does not show any clear improvement for the peer instruction classes.

Table 4.6: Student survey on peer instruction implementation

From the point of helping me learn, the content of clicker questions was				
Much too hard 0%	Too hard 6.66%	OK 80%	Too easy 13.33%	Much too easy 0%
In general, the instructor gave us enough time to read and understand the questions before the first vote.				
No, far too little time 0%	No, too little time 0%	OK amount of time 80%	Yes, too much time 13.33%	Yes, far too much time 6.66%
Which of the following best describes your discussion practices in this group?				
I always discuss with the group around me, it helps me learn 66.66%	I always discuss with the group around me, I don't really learn, but I stay awake 10%	I sometimes discuss, it depends 22.33%	I rarely discuss, I don't think I get a lot out of it 0%	I rarely discuss, I'm too shy 0%
The amount of time generally allowed for peer discussion was				
Much too short 3.33%	Too short 11%	About right 89%	Too long 0%	Much too long 0%
In general, the time allowed for class-wide discussion (after the group vote) was				
Much too short 0%	Too short 6.66%	About right 70%	Too long 23.33%	Much too long 0%
In general, it was helpful for the instructor to begin class-wide discussion by having students give an explanation.				
N/A - The instructor rarely did this 16.66%	It's not helpful to hear other students' explanations 10%	It was helpful to hear other students' explanations 73.33%		
The professor explained the value of using clickers in this class.				
Not at all 0%	Somewhat, but I was still unclear why we were doing it 10%	Yes, they explained it well 83.33%	Yes, they explained it too much 6.66%	

4.3.2 Failure Rate

To measure the students' performance in the course for both traditional and peer instruction classes, we obtain failure rate in quizzes and subjective exams. The university policy defines that the passing grades are `A`, `B`, and `C` and the failing grades are `D`, and `F`. If a student scores less than 70% marks, he/she will be considered failed.

Class Quiz Figure 4.3 presents the failure rate of undergraduate and graduate students in quiz exams for both traditional-lecture (Fall 2015 and Fall 2016) and peer instruction (Fall 2017) classes. The results show noticeable improvements in the failure rates for peer instruction. In particular, failure rate of graduate students in user authentication topic is reduced to zero in Fall 2017 from

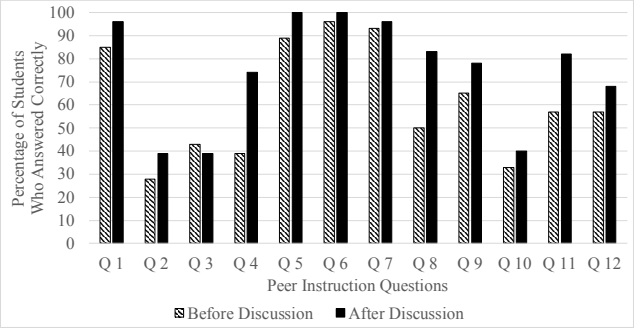


Figure 4.5: Percentage of the students who respond to the peer instruction questions correctly

18% and 33% in Fall 2015 and Fall 2016 respectively. The undergraduate students have performed significantly better in the quiz on cryptographic tools when peer instruction is used. Their failure rate is reduced by 44% and 37% as compared to Fall 2015 and Fall 2016 respectively.

Subjective Exam Figure 4.4 shows the failure rate of the subjective exams for the five topics (i.e., computer security overview, buffer overflow, user authentication, malicious software, and cryptographic tools) taught at both traditional-lecture and peer instruction classes. We notice substantial improvements in the failure rate for undergraduate students when peer instruction is used except the cryptographic tools. We reevaluated the student answers of the questions on this topic. In particular, we found that a significant number of students misunderstood the following question.

Question on Cryptographic Tools: `How can message authentication be achieved using one-way hash function with 1) Symmetric encryption and 2) Public-key encryption.`

Apparently, they ignore the one-way hash function and assume that the question asks about the symmetric and Public-key encryption schemes. Some students derive message authentication through encryption without computing and utilizing cryptographic hash values. If the question is rephrased and restructured, it will likely reduce the failure rate on this topic.

The failure rate for graduate students do not show any clear trend. Overall, we notice that the peer instruction reduces the failure rate when compared with the traditional lecture classes in Fall 2016. However, it shows no improvement when compared with Fall 2015.

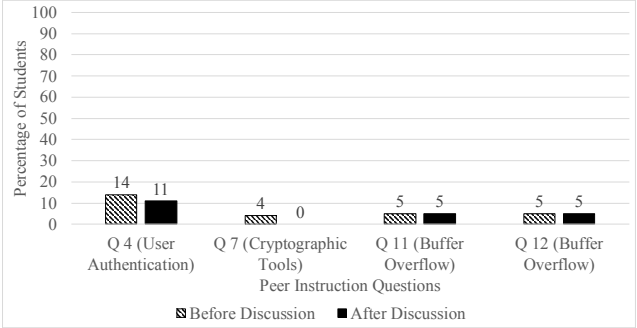


Figure 4.6: Percentage of the students who respond to the peer instruction questions with the clicker choices that are not given in the questions

4.3.3 Learning Gain during Group Discussions

Figure 4.5 presents the results of the clicker responses of the students before and after the group discussions. We notice clear evidence of improvement in the correct answers by the students after the discussions.

To our surprise, some students chose an option from clickers that were not given in the questions. In particular, we observed these choices in the questions 4, 7, 11, and 12 on three topics: user authentication, cryptographic tools, and buffer overflow. Figure 4.6 shows the percentage of the students who have selected unexpected clicker options. Following is an example of such question.

Question on Buffer Overflow: `Which of the following describes a buffer overflow attack?`

1. Exploiting the traffic flow mechanism in a buffer and blocking packets from reaching their destination.

2. Flooding a buffer with server requests and overflowing the network bandwidth.

3. Attempting to store more input in a data holding area than capacity allocates.

4. An attacker fills the target buffer with malicious code

The above question has four choices: `A`, `B`, `C`, and `D`. However, some students respond with `E` from clickers. It shows that these students do not pay attention to the questions. We also notice in Figure 4.6 that some of these students change their responses after the group discussions, depicting that they start paying attention during the discussions.

4.3.4 Survey

Table 4.5 presents the results of the student attitudinal survey portion of the peer instruction evaluation. It shows that the most of the students find it useful to think about a clicker question before discussing it with other students and the discussion helps them understand the concept better. 70% of students would recommend peer instruction be adopted by other instructors.

Table 4.6 summarizes the students opinion about the peer instruction classes. It shows that the students have a generally positive experience of the classes. They have adequate time to understand the questions and vote for the correct answer. 80% students agree that the allowable duration for group discussions is sufficient.

Chapter 5

Conclusion

5.1 Assessment of Concept Maps

The document presented 41 concept maps useful for improving learning experience of students in class. We evaluated the quality of the concept maps using two well known techniques for analysis the structure and contents of concept maps. A well-defined rubric with six elements and four levels of quality i.e. excellent, good, poor, fail for content. Topological taxonomy features where the taxonomy score indicates the level of concept map i.e, higher the taxonomy score greater the structure of concept maps. The evaluation results show that for SCADA security concept maps out of 22 concept maps 37.12% of Cmaps are of excellent quality and 52.27% of concept maps are good. For digital forensics concept maps out 19 concept maps 42.9% of CMaps are of excellent quality and 50% of concept maps are good. From topological taxonomy features analysis, for SCADA security, out of 22 maps 10 concept maps have above average taxonomy score where as for digital forensics investigation out of 19 maps 9 maps score above average rank.

5.2 Evaluation of Peer Instruction

We implemented and evaluated peer instruction in a semester-long course, introduction to computer security. The evaluation results were compared with traditional lecture classes in terms of dropout rate, failure rate, and student learning gain. Peer instruction showed promising results for undergraduate students. Their dropout rate was reduced by by 6% and 16% and failure rate by 44% and 37% when compared with traditional lecture classes of two semesters respectively. The survey results showed that 77% students found the discussions in small groups useful to understand the computer security concepts. The overall student experience of peer instruction was positive and majority students would recommend peer instruction be adopted by other instructors. Unfor-

tunately, the impact of peer instruction on graduate students is not apparent in evaluation results. Generally, the existing efforts on peer instruction mostly focused on undergraduate curriculum. At large, graduate-student is an unknown factor in peer instruction. We suggest and encourage computing education community to implement and evaluate peer instruction for graduate classes.

Bibliography

[1] Cmapanalysis: an extensible concept map analysis tool. https://github.com/lbunch/cmapanalysis.

[2] Rubric for assessing concept maps (centre for teaching excellence, university of waterloo). https://uwaterloo.ca/centre-for-teaching-excellence/sites/ca.centre-for-teaching-excellence/files/uploads/files/rubric_for_assessing_concept_maps.pdf, 2016.

[3] I. Ahmed, S. Obermeier, M. Naedele, and G. G. Richard III. SCADA Systems: Challenges for Forensic Investigators. *Computer*, 45(12):44–51, Dec 2012.

[4] I. Ahmed, S. Obermeier, S. Sudhakaran, and V. Roussev. Programmable Logic Controller Forensics. *IEEE Security Privacy*, 15(6):18–24, November 2017.

[5] I Ahmed, V Roussev, W Johnson, S Senthivel, and S Sudhakaran. A SCADA System Testbed for Cybersecurity and Forensic Research and Pedagogy. In *Proceedings of the 2nd Annual Industrial Control System Security Workshop*, ICSS '16, pages 1–9, New York, NY, USA, 2016. ACM.

[6] Irfan Ahmed and Vassil Roussev. Peer instruction teaching methodology for cybersecurity education. *IEEE Security Privacy*, 16(4), July 2018.

[7] Manish Bhatt, Irfan Ahmed, and Zhiqiang Lin. Using virtual machine introspection for operating systems security education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 396–401. ACM, 2018.

[8] Alberto J Cañas, Larry Bunch, Joseph D Novak, and Priit Reiska. Cmapanalysis: An extensible concept map analysis tool. *Journal for Educators, Teachers and Trainers*, 2013.

[9] Alberto J Cañas and Joseph D Novak. Concept mapping using cmaptools to enhance meaningful learning. In *Knowledge cartography*, pages 25–46. Springer, 2008.

[10] Ronald N Cortright, Heidi L Collins, and Stephen E DiCarlo. Peer instruction enhanced meaningful learning: ability to solve novel problems. *Advances in physiology education*, 29(2):107–111, 2005.

[11] Catherine H Crouch and Eric Mazur. Peer instruction: Ten years of experience and results. *American journal of physics*, 69(9):970–977, 2001.

[12] Pranita Deshpande. *Concept Map Datasets for Cybersecurity Courses*, 2018 (accessed July 23, 2018).

[13] Pranita Deshpande and Irfan Ahmed. Topological scoring of concept maps for cybersecurity education. In *50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*, SIGCSE '19, Minneapolis, Minnesota, USA, 2019. ACM.

[14] Pranita Deshpande, Irfan Ahmed, and Cynthia B. Lee. Evaluation of peer instruction for cybersecurity education. In *50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*, SIGCSE '19, Minneapolis, Minnesota, USA, 2019. ACM.

[15] Sarah Esper. A discussion on adopting peer instruction in a course focused on risk management. *J. Comput. Sci. Coll.*, 29(4):175–182, April 2014.

[16] William Johnson, Irfan Ahmed, Vassil Roussev, and Cynthia B. Lee. Peer instruction for digital forensics. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*, Vancouver, BC, 2017. USENIX Association.

[17] William E. Johnson, Allison Luzader, Irfan Ahmed, Vassil Roussev, Golden G. Richard III, and Cynthia B. Lee. Development of peer instruction questions for cybersecurity education. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, Austin, TX, 2016. USENIX Association.

[18] Nishchal Kush, Ernest Foo, Ejaz Ahmed, Irfan Ahmed, and Andrew Clark. Gap analysis of intrusion detection in smart grids. 01 2011.

[19] Cynthia Bailey Lee, Saturnino Garcia, and Leo Porter. Can peer instruction be effective in upper-division computer science courses? *Trans. Comput. Educ.*, 13(3):12:1–12:22, August 2013.

[20] Chen-Chung Liu, Ping-Hsing Don, Chun-Ming Tsai, et al. Assessment based on linkage patterns in concept maps. *Journal of information science and engineering*, 21(5):873–890, 2005.

[21] Norma L Miller and Alberto J Cañas. A semantic scoring rubric for concept maps: design and reliability. 2008.

[22] Joseph D Novak and Alberto J Cañas. The origins of the concept mapping tool and the continuing evolution of the tool. *Information visualization*, 5(3):175–184, 2006.

[23] Joseph D Novak and D Bob Gowin. *Learning how to learn*. Cambridge University Press, 1984.

[24] Joseph D Novak and Dismas Musonda. A twelve-year longitudinal study of science concept learning. *American educational research journal*, 28(1):117–153, 1991.

[25] Leo Porter, Cynthia Bailey Lee, and Beth Simon. Halving fail rates using peer instruction: a study of four computer science courses. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 177–182. ACM, 2013.

[26] Leo Porter, Dennis Bouvier, Quintin Cutts, Scott Grissom, Cynthia Lee, Robert McCartney, Daniel Zingaro, and Beth Simon. A multi-institutional study of peer instruction in introductory computing. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, pages 358–363, New York, NY, USA, 2016. ACM.

[27] Leo Porter, Saturnino Garcia, John Glick, Andrew Matusiewicz, and Cynthia Taylor. Peer instruction in computer science at small liberal arts colleges. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '13, pages 129–134, New York, NY, USA, 2013. ACM.

[28] Sumangala P Rao and Stephen E DiCarlo. Peer instruction improves performance on quizzes. *Advances in Physiology Education*, 24(1):51–55, 2000.

[29] Vassil Roussev. Digital forensic science: issues, methods, and challenges. *Synthesis Lectures on Information Security, Privacy, & Trust*, 8(5):1–155, 2016.

[30] S Senthivel, I Ahmed, and V Roussev. SCADA network forensics of the PCCC protocol. *Digital Investigation*, 22:S57–S65, 2017.

[31] Saranyan Senthivel, Shrey Dhungana, Hyunguk Yoo, Irfan Ahmed, and Vassil Roussev. Denial of Engineering Operations Attacks in Industrial Control Systems. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, CODASPY '18, pages 319–329, New York, NY, USA, 2018. ACM.

[32] Beth Simon and Quintin Cutts. Peer instruction: A teaching method to foster deep understanding. *Commun. ACM*, 55(2):27–29, February 2012.

[33] Beth Simon, Michael Kohanfars, Jeff Lee, Karen Tamayo, and Quintin Cutts. Experience report: Peer instruction in introductory computing. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE '10, pages 341–345, New York, NY, USA, 2010. ACM.

[34] Alejandro Valerio, David B Leake, and Alberto J Cañas. Automatic classification of concept maps based on a topological taxonomy and its application to studying features of human-built maps. 2008.

[35] Daniel Zingaro, Cynthia Bailey Lee, John Glick, Leo Porter, and Beth Simon. Peer instruction in cs: Introduction and recent developments (abstract only). In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 764–764, New York, NY, USA, 2013. ACM.

Vita

Pranita Deshpande received her Bachelor Degree in Engineering from Basaveshawara Engineering college , India in 2015. She joined the University of New Orleans for Computer Science Master of Science program in Spring 2017. She started working as a Research Assistant in Cy-Phy Laboratory at University of New Orleans under the supervision of Dr. Irfan Ahmed. She have been doing active research in cybersecurity education improvement for courses like SCADA security, Digital Forensics, computer security. Developing and evaluating the well defined pedagogical tools used in improvised learning curriculum.